```
entry:
                                         %result.addr = alloca ptr, align 8
                                         %input1.addr = alloca ptr, align 8
                                         %input2.addr = alloca ptr, align 8
                                         %size.addr = alloca i32, align 4
                                         %r = alloca [4 x i8], align 1
                                         \%i = alloca i32, align 4
                                         \%i7 = alloca i32, align 4
                                         store ptr %result, ptr %result.addr, align 8
                                         store ptr %input1, ptr %input1.addr, align 8
                                         store ptr %input2, ptr %input2.addr, align 8
                                         store i32 %size, ptr %size.addr, align 4
                                         call void @llvm.memcpy.p0.p0.i64(ptr align 1 %r, ptr align 1
                                        ... @ const.public comp.r, i64 4, i1 false)
                                         store i32 0, ptr %i, align 4
                                         br label %for.cond
                                                 for.cond:
                                                 %0 = load i32, ptr %i, align 4
                                                 %1 = load i32, ptr %size.addr, align 4
                                                 %cmp = icmp slt i32 %0, %1
                                                  br i1 %cmp, label %for.body, label %for.end
for.body:
%2 = load ptr, ptr %input1.addr, align 8
%3 = load i32, ptr %i, align 4
%idxprom = sext i32 \%3 to i64
%arrayidx = getelementptr inbounds i8, ptr %2, i64 %idxprom
%4 = load i8, ptr %arrayidx, align 1
%conv = sext i8 %4 to i32
%5 = load ptr, ptr %input2.addr, align 8
%6 = load i32, ptr %i, align 4
%idxprom1 = sext i32 \%6 to i64
                                                                            for.end:
                                                                            store i32 0, ptr %i7, align 4
%arrayidx2 = getelementptr inbounds i8, ptr %5, i64 %idxprom1
%7 = load i8, ptr %arrayidx2, align 1
                                                                            br label %for.cond8
%conv3 = sext i8 \%7 to i32
%mul = mul nsw i32 %conv, %conv3
%conv4 = trunc i32 %mul to i8
%8 = load ptr, ptr %result.addr, align 8
%9 = load i32, ptr %i, align 4
%idxprom5 = sext i32 \%9 to i64
%arrayidx6 = getelementptr inbounds i8, ptr %8, i64 %idxprom5
store i8 %conv4, ptr %arrayidx6, align 1
br label %for.inc
                                                                 for.cond8:
                      for.inc:
                                                                 %11 = load i32, ptr %i7, align 4
                      %10 = load i32, ptr %i, align 4
                                                                 %12 = load i32, ptr %size.addr, align 4
                      %inc = add nsw i32 %10, 1
                                                                 %cmp9 = icmp slt i32 %11, %12
                      store i32 %inc, ptr %i, align 4
                                                                 br i1 %cmp9, label %for.body11, label %for.end21
                      br label %for.cond, !llvm.loop !6
       for.body11:
       %13 = load i32, ptr %i7, align 4
       %idxprom12 = sext i32 %13 to i64
       %arrayidx13 = getelementptr inbounds [4 x i8], ptr %r, i64 0, i64 %idxprom12
       %14 = load i8, ptr %arrayidx13, align 1
       %conv14 = sext i8 %14 to i32
       %15 = load ptr, ptr %result.addr, align 8
       %16 = load i32, ptr %i7, align 4
                                                                                                  for.end21:
       %idxprom15 = sext i32 %16 to i64
                                                                                                  ret void
       %arrayidx16 = getelementptr inbounds i8, ptr %15, i64 %idxprom15
       %17 = load i8, ptr %arrayidx16, align 1
       %conv17 = sext i8 %17 to i32
       %add = add nsw i32 %conv17, %conv14
       %conv18 = trunc i32 %add to i8
       store i8 %conv18, ptr %arrayidx16, align 1
       br label %for.inc19
                                                    for.inc19:
                                                    %18 = load i32, ptr %i7, align 4
                                                    \%inc20 = add nsw i32 %18, 1
                                                    store i32 %inc20, ptr %i7, align 4
                                                    br label %for.cond8, !llvm.loop !8
```

CFG for 'public comp' function